

Ivan Iwannado

Ivan Iwannado knows what he wants to do, but like most of us he'll take a while to figure out how to do it. Learn from his mistakes



THIS MONTH:

IVAN CREATES AN ELECTRONIC NEWSLETTER

Ivan Iwannado has decided to create an electronic newsletter to advertise the amazing events that are taking place at his village hall this year. He's been trying to work out the best way to offer this to his eager public. He wants something that he can email to everyone who has been daft enough to add their name to the parish mailing list. It needs to look good and everyone has to be able to read it. But he's not at all sure how to go about making this publication. It all seems rather involved, and Ivan is beginning to think maybe he should just make a traditional newsletter using a trusty photocopier and post it in the old-fashioned way.

Ivan should stick with his original idea. Electronic newsletters are cheaper to produce, easier to deliver and look much better.

Can't Ivan just create documents in Microsoft Office? Most people seem to own that.

Not really. For one thing, lots of people don't use Microsoft Office. Ivan's going to annoy people if he sends a document in Word format to someone who doesn't have Word. For another thing, Word isn't that brilliant at making clever layouts.

What is a generally supported format then?

The only completely common format is plain text: no fonts, no pictures, just the writing.

Well, that won't look very classy. Even the old photocopier was better than that! Come on, Ivan wants pretty pictures and lots of colour.

The professional option is to use Adobe's portable document format, commonly known as PDF. This file format has been the market leader for many years and with good reason. It has great layout controls and can look incredibly attractive. Furthermore, anyone who has the free Acrobat software will be able to read files saved in this format. Most computers have Adobe Acrobat Reader installed. And if someone doesn't have it, they can simply download it from the web.

What does Ivan need to create PDFs?

He could get the full version of the Acrobat software, though he will have to pay for it.



What, spend money? Isn't there an alternative format that Ivan could use?

You can use free utilities for creating PDFs. CutePDF Writer is one such program and it is available from www.cutepdf.com. If making PDFs sounds too scary, then Ivan could consider the rich text format (RTF), a Microsoft proprietary format. He can save a Word document as an RTF; this is more widely compatible than the .doc format.

What else is out there?

Most people have a web browser so they can read hypertext markup language (HTML) files. However, these files are not ideal when you need to make complex layouts.

Why? The whole web is done in HTML, isn't it?

Yes, but the site that Ivan looks at on his computer may look quite different on another person's PC, with different fonts and so on. Anyway, Ivan still needs software to be able to create his HTML file.

So if Ivan wanted to create HTML files, what would he use?

All the big-name word processors have the option of saving their documents in HTML format. Then there are all the specialist web design packages that he can

use to create non-web documents as well as web pages. There are also many free HTML editors around. However, if Ivan's readers' computers don't have the fonts that Ivan has used on his pages, they'll have problems seeing the pages exactly as they were intended. HTML doesn't embed fonts, so if a particular font is missing, the web browser just has to use the nearest possible match.

Is there any way that Ivan can make sure his files will display correctly on most computers?

If Ivan sticks to the most common fonts he'll have a better chance than if he uses an obscure font that no-one else has ever heard of, much less installed on their computer. It's also sensible to assume that some people will have smaller monitors with lower screen resolutions. Designing pages for a screen that can display 800x600 pixels will mean that they have a strong chance of displaying properly on everyone's screen. But designing layouts for a display that uses a very high resolution will cause problems for many computer users.

Having thought about all the options, Ivan has decided to opt for the simplicity of Adobe PDF files. That way, more people will be able to view his booklet and they'll also be able to print it out and stick it on their notice board. He doesn't want to pay for a full copy of Adobe Acrobat, though, so he'll try a few of the freebies first. 

Nigel Knowitall



If you want to get something done using software but are unsure how to do it, fear not — help is at hand from our resident expert, **Nigel Knowitall**

EMAILING A WORD DOCUMENT AUTOMATICALLY

Q In the February 2005 issue of *Nigel Knowitall*, you answered a query about emailing an Excel spreadsheet automatically. This has proved very useful to me. Can the same be done with a Word document?

Each day I receive between 40 and 50 emails from different people. Each has an attached Word document containing a table. After finding the information requested, I complete the table and type my name at the bottom. To return the Word document I go to File, Send To, Mail Recipient and so on. However, all the documents go to the same contact. I'm trying to devise a way to send them by pressing a key sequence after I've entered my name. This

should send the document to the central contact automatically. How do I set this up?
Peter Horrocks

A You might think that the easiest way to do this would be to create a macro that reads as follows:

```
Sub senddoc()  
ActiveDocument.SendMail  
Recipients:="Ivan Iwannado"  
End Sub
```

The problem is that, unlike Excel, Word's SendMail method doesn't let you set a recipients property. There are two methods that you can use instead. The easiest is Word's RoutingSlip method, which lets you send a document to one or more

recipients. You can set the subject for the message and choose whether it is sent to all the recipients at once or routed from one to the next.

The problem with RoutingSlip is that you can't change the body text of the email message. Also, you can only send the document as an attachment and the document you are working on is changed to be protected with Track Changes. A final potential problem is that you cannot put any recipients in the Bcc field.

Other than that, RoutingSlip is easy to use. Open the Visual Basic Editor using Tools, Macro, Visual Basic Editor and create a new procedure with the following code:

```
Sub senddoc()  
ActiveDocument.HasRoutingSlip = True  
With ActiveDocument.RoutingSlip
```

MANAGING VALIDATION CONDITIONS

Q I've inherited an Excel spreadsheet that lets me input only specific values into various cells. It's a great idea when it works, but I find it frustrating when it blocks me from adding perfectly valid entries. So there are two things I'd like to find out. First, how can I add the same facility to other spreadsheets that I manage? Second, how do I add to, change or remove the values that I can input to the spreadsheet that I've inherited?

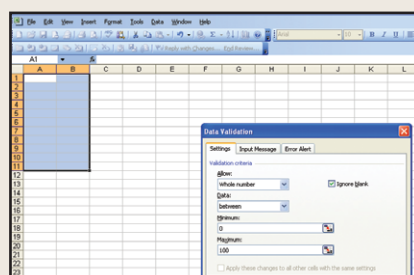
Lisa Maran

A Excel lets you restrict the data that can be entered in cells to specific values by using validation conditions. We'll start by adding validation conditions to a new workbook. Select an area of the workbook where you want to restrict the data entered to specific values, then open the Data, Validation menu. This brings up a dialog box that lets you specify the values that are allowed in the selected area.

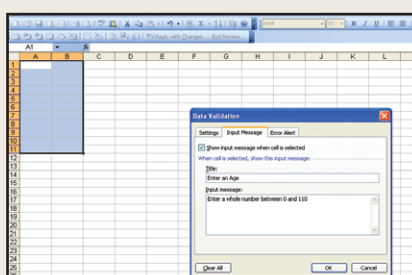
The default choice is to allow any value, which is what happens if you don't set anything up.

However, there are several things you can do to make data entry more reliable. First, you can set the cells to accept only whole numbers. This is useful if your users are entering ages in years, for example.

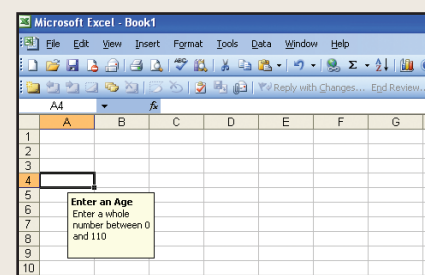
Click the Settings tab and select 'Whole number' from the Allow list. You can set various choices about the whole numbers you want to accept, such as numbers in a given range, numbers less than a specific value and so on. So in our age example, you could choose Between and have a minimum of 0 and a maximum of 110, say.



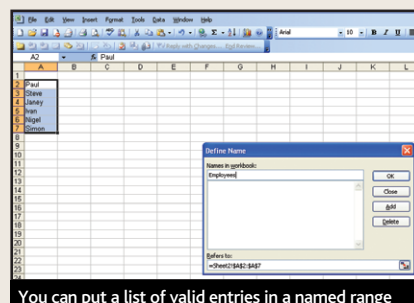
You can set up data validation criteria to limit the data that can be entered into cells



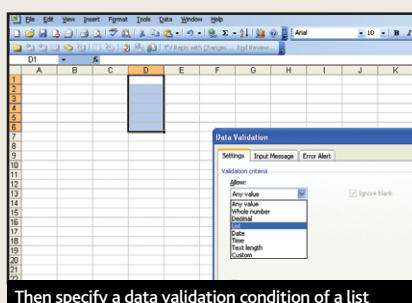
Remember to enter a suitable message telling the user what they should type



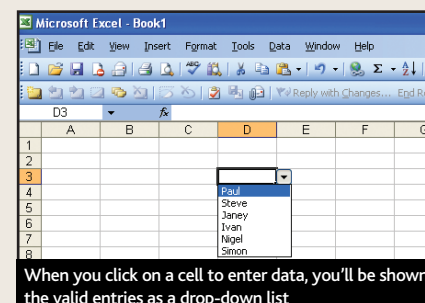
When you click on one of the cells in the area, you will see a message that tells you what to enter



You can put a list of valid entries in a named range



Then specify a data validation condition of a list



When you click on a cell to enter data, you'll be shown the valid entries as a drop-down list



```
.Subject = "New subject goes
here"
.AddRecipient "Ivan Iwannado"
End With
Activatedocument.Route
End Sub
```

Replace the bit that reads "Ivan Iwannado" with the name of someone in your contact list, or with an actual email address along the lines of fred.bloggs@mycompany.co.uk.

If you then assign this macro to a key combination, you'll be able to send the document by pressing the key combination. Microsoft is so paranoid about viruses that propagate by sending emails automatically, the chances are that when you start the macro you'll see a message that reads, "A program is trying to automatically send an email on your behalf. Do you want to allow this?" You'll have to press the Yes button twice before the email is sent.

This will appear if you've installed the security patch for Outlook 2000 or if you are running Outlook 2002 or 2003. One way around this is to download the Express ClickYes utility. This is available free from www.express-soft.com/mailmate/clickyes.html. It sits in the taskbar and clicks the Yes button on your behalf when Outlook asks the question.

Microsoft made the changes to Outlook's security so it brings up this message to prevent

viruses sending emails without the user of the PC knowing; it seems a bit perverse to disable the security by bypassing it in this way. It would be more sensible if Outlook had an option for the authorised user to have a list of macros that could be used to send emails. I don't know of any way to do this yet, unfortunately. If any readers know of a solution to this problem, please let us know.

To avoid the problems mentioned with the RoutingSlip method, you could try automating Outlook instead. You'll still see the annoying warning message, but at least your document isn't set to be protected with Track Changes. For this macro to work, you need to have saved your document and you must add a reference in your Visual Basic for Applications (VBA) Editor to the appropriate Outlook library. From the Tools menu in VBA, choose References. Look for Microsoft Outlook 11 Object Library, or Outlook 10 or 9 – whatever your version happens to be:

```
Sub SendDoc2()
    Dim bStarted As Boolean
    Dim oOutlookApp As Outlook.Application
    Dim oItem As Outlook.MailItem
    On Error Resume Next
    Set oOutlookApp = GetObject
```

```
(, "Outlook.Application")
If Err <> 0 Then
    Set oOutlookApp = CreateObject("Outlook.Application")
    bStarted = True
End If
Set oItem = oOutlookApp.CreateItem(oMailItem)
With oItem
    .To = "fred.bloggs@mycompany.co.uk"
    .Subject = "Shopper Test"
    .Attachments.Add Source:=ActiveDocument.FullName, Type:=olByValue, _
    DisplayName:="Document as attachment"
    .Send
End With
If bStarted Then
    oOutlookApp.Quit
End If
Set oItem = Nothing
Set oOutlookApp = Nothing
End Sub
```

This macro starts by checking whether Outlook is already running using the GetObject command. If it isn't, this generates an error and Outlook is then started by the part of the code that reads:

```
If Err <> 0 Then
    Set oOutlookApp = CreateObject("Outlook.Application")
    bStarted = True
End If
```

The mail item is created and various values are set such as the recipient (.To), the subject line and the attachment information. Once the mail message has been sent, the macro checks whether it started Outlook. If it did, it closes Outlook again.

MAINTAINING FORMATTING

Q Whenever I insert data into my Excel workbooks from other sources it keeps the formatting from the original source. This is irritating in workbooks where I've set up the font, size and colour in the tables to make different areas clear. Is there any way to make incoming data assume the format of the cell into which it has been inserted?

Jason Hunt

A All you have to do is double-click on the cell before you paste in the content and the data will use the formatting of the workbook rather than the originating application. **ES**

CONTACT

NIGEL KNOWITALL

If you want to get something done with software ask the expert, Nigel Knowitall

nigel@computershopper.co.uk

When you set a data validation range, be sensible and give the user messages that tell them what they are supposed to enter into the cells; making them guess what is acceptable is annoying. Fill in the Input Message tab with a message containing clues about valid entries. This pops up in a neat little box whenever the user selects a cell that has the data validation rule. You can also set an error alert that appears if the user enters a value that doesn't match the validation conditions. If this happens, the user is shown the message and given a chance to put things right. You're not just limited to specifying ranges of whole numbers, of course. You can give ranges of decimal numbers, dates and times or specify a maximum or minimum text length.

From the description that you've given, it sounds as though the person from whom you inherited your workbook specified a list of valid entries. The setting for this is List. When you want to have a list of valid entries, you can choose an area or predefined named range on your worksheet that contains the list of valid entries. For example, you could have a list of valid employee names in the range K1:K10. You could then set up a data entry area in your worksheet where users can enter information including the employee name.

If you give this a Data Validation setting of List, you'll be given the chance to enter the range or range name of cells that contain the list. The great thing about this is that the lists of valid entries don't have to be on the same sheet in the workbook as the data entry area. You could set up the list of employee names on Sheet 2, and as long as you have defined that range with a suitable

name, you can then refer to it on Sheet 1. To try this out on your sample workbook, go to Sheet 2 and pick an area to hold your data. We'll use column A and set the data range to be A2:A10. Enter a set of employee names into that range and specify its name using Insert, Name, Define. If you have selected the range containing the data before doing this, you are given that area as the default range. Specify the range name as Employees.

Now you can go back to Sheet 1 of the workbook. Pick an area where you would like to be able to enter names from the employee name list. I've used the range D1 to D6. Select that range, then choose Data, Validation. Then select List from the options you are offered. In the Source box, type =Employees.

When the user chooses a cell in the data entry range, they will be shown a drop-down list containing those data values. Clicking on one of the values will place the entry into the cell. Alternatively, they can type a value; as long as it is contained in the list, it will be accepted.

If you are having problems with the spreadsheet you inherited because you can't enter all the values you would like to enter, you will need to alter the values in the named range or area of the spreadsheet to which the data validation setting refers. You can either change some of the existing values or extend the range and add the new values.

Finally, you can remove data validation from an area of your spreadsheet. To remove all validation from a particular area, select the cells in question and then from the Data menu click Validation followed by Clear All. The validation conditions will be removed.